Mit WeDo 2.0 Kompetenzen des informatischen Denkens fördern

In diesem Kapitel erfahren Sie, wie Sie WeDo 2.0 einsetzen, um Fähigkeiten des informatischen Denkens in einem naturwissenschaftlichen Kontext zu fördern.

Mit LEGO® Education WeDo 2.0 Kompetenzen des informatischen Denkens fördern

LEGO® Education freut sich, Ihnen diese Projekte vorstellen zu dürfen, die speziell für den Einsatz im Grundschulunterricht entwickelt wurden. Sie sollen den Schülerinnen und Schülern dabei helfen, ihre Kompetenzen im Bereich informatisches Denken zu entwickeln.

Informatisches Denken umfasst eine Reihe von Fähigkeiten, die zum Lösen alltäglicher Probleme wichtig sind. Bei WeDo 2.0 werden diese Fähigkeiten in den einzelnen Phasen eines Projekts schrittweise vermittelt. In jedem Projekt haben Sie die Möglichkeit, verschiedene Kompetenzen zu fördern. Sie können jeweils selbst entscheiden, welche davon für Sie und Ihre Klasse relevant sind, und sich speziell darauf konzentrieren.

Bei jedem Projekt von WeDo 2.0 kommen sowohl LEGO® Steine als auch eine symbolbasierte Programmiersprache zum Einsatz. Auf diese Weise erhalten die Schülerinnen und Schüler beim Lösen der Fragestellungen eine Einführung in die Grundprinzipien des Programmierens.

WeDo 2.0 fördert das informatische Denken durch Aufgaben zum Programmieren. Dadurch werden die Kreationen der Schülerinnen und Schüler zum Leben erweckt. Das stellt eine großartige Lernerfahrung dar und regt sie dazu an, mehr entdecken zu wollen.

Informatik, informatisches Denken, programmieren

Naturwissenschaft und Technik sind sehr alte Disziplinen in der menschlichen Geschichte. Im Gegensatz dazu ist die Informatik noch sehr jung. Trotzdem hat sie in kürzester Zeit nicht nur unsere Herangehensweise an die Naturwissenschaft und Technik verändert, sondern auch unsere grundlegende Lebensweise.

Die Informatik gehört zu den MINT-Fächern und ähnelt somit den Disziplinen Mathematik, Naturwissenschaft und Technik.

Alle MINT-Fächer eignen sich hervorragend dafür, Denkweisen und Kompetenzen zu entwickeln, von denen die Schülerinnen und Schüler ein Leben lang profitieren. Dazu gehören beispielsweise das Finden von Fragestellungen, das Entwickeln von Lösungen und das Präsentieren von Ergebnissen.

Auch das informatische Denken gehört zu diesen Fähigkeiten. Es beschreibt eine Art zu denken und Probleme zu lösen.

Das informatische Denken umfasst eine Reihe von Fähigkeiten, zu der auch das Programmieren und das algorithmische Denken gehört.

Damit stellt das Programmieren eine Möglichkeit dafür dar, das algorithmische Denken im Kontext der MINT-Fächer zu erlernen.

MINT-Fächer

Mathematik, Informatik,
Naturwissenschaften, Technik

Denkweisen und Kompetenzen entwickeln, von denen Schülerinnen und Schüler ein Leben lang profitieren

1. Fragen stellen und Probleme lösen
2. Modelle einsetzen
3. Prototypen entwickeln
4. Forschendes Lernen
5. Daten analysieren und interpretieren
6. Informatisches Denken anwenden
   a. Zerlegen
   b. Abstrahieren
   c. Algorithmisch denken
   d. Evaluieren
   e. Generalisieren
7. Auf Grundlage von Nachweisen argumentieren
8. Informationen zusammentragen, auswerten und anderen mitteilen

Was bedeutet informatisches Denken?

Der Begriff „informatisches Denken” wurde von Seymour Papert eingeführt und von der Professorin Jeannette Wing bekannt gemacht. Sie definierte den Begriff folgendermaßen:

„Der Denkprozess, der stattfindet, wenn man Fragestellungen und die dazugehörigen Lösungen formuliert. Der Lösungsweg wird dabei in einer Form dargestellt, die sich von einem informationsverarbeitenden Agenten effektiv durchführen lässt.“ (Wing, 2011)

Informatisches Denken kommt in zahlreichen Bereichen und Situationen zum Einsatz. Tatsächlich nutzen wir es sogar im alltäglichen Leben. Die Kompetenzen des algorithmischen Denkens spielen in der Naturwissenschaft, Technik und Mathematik eine Rolle. Diese Kompetenzen können folgendermaßen beschrieben werden:

Zerlegung

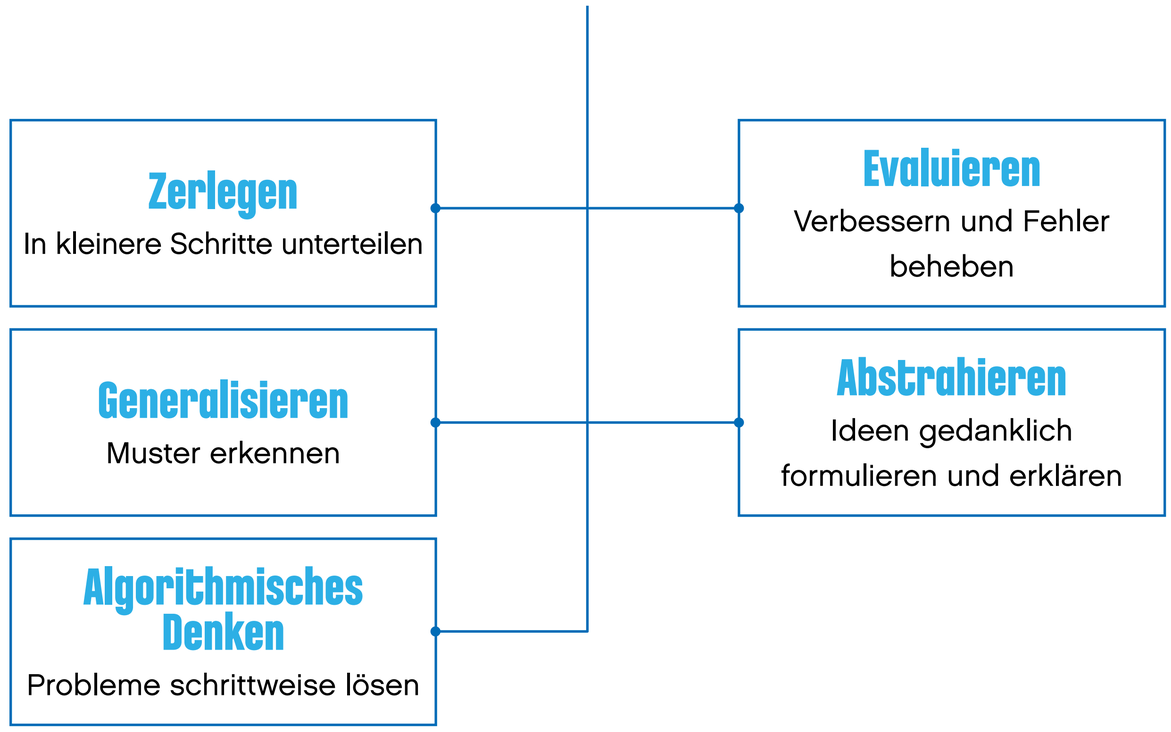
Zerlegung beschreibt die Fähigkeit, eine Aufgabe in kleinere Schritte zu unterteilen, um den Lösungsfindungsprozess zu vereinfachen. Auf diese Weise lässt sich das Problem anderen einfacher erklären und in einzelne Schritte unterteilen. Auf die Zerlegung folgt häufig die Verallgemeinerung.

Beispiel: Wenn man in den Urlaub fährt, kann die Vorbereitung (oder das Projekt) in kleinere Aufgaben unterteilt werden: Flüge buchen, Hotel reservieren, Koffer packen usw.

Generalisierung (Erkennen von Mustern)

Verallgemeinerung beschreibt die Fähigkeit, Bestandteile einer Aufgabe zu erkennen, die bereits bekannt sind oder die man schon in einem anderen Zusammenhang kennengelernt hat. Damit lassen sich in vielen Fällen einfachere Möglichkeiten dafür finden, einen Algorithmus zu entwickeln.

Beispiel: Die Funktionsweise von Ampeln besteht darin, die gleiche Abfolge von Signalen unendlich zu wiederholen.



Was bedeutet informatisches Denken?

Informatisches Denken

Das informatische Denken ist die Fähigkeit, eine geordnete Abfolge von Schritten zu erstellen, um ein Problem zu lösen.

Beispiel 1: Bei einem Kochrezept befolgen wir eine Reihe von Schritten, um eine Mahlzeit zuzubereiten.
Beispiel 2: Beim Spielen mit Computern können wir eine Abfolge von Befehlen kodieren, die dem Computer mitteilen, was er tun soll.

Evaluieren oder Fehler beheben

Hierbei handelt es sich zum einen um die Fähigkeit, zu beurteilen, ob ein Prototyp wie erwartet funktioniert oder nicht. Zum anderen gehört dazu auch die Fähigkeit, zu bestimmen, wo Verbesserungen erforderlich sind, falls der Prototyp nicht wie erwartet funktioniert. Es ist der gleiche Prozess, den ein Computerprogrammierer durchläuft, um Fehler in einem Programm zu finden und zu beheben.

Beispiel 1: Beim Kochen probieren wir regelmäßig das Essen, um festzustellen, ob es richtig gewürzt ist oder nicht.
Beispiel 2: Wenn wir einen von uns geschriebenen Text auf Tippfehler und fehlende Satzzeichen überprüfen, führen wir eine Fehlersuche und -behebung durch, damit die Leser den Text gut verstehen können.

Abstraktion

Abstraktion bezeichnet die Fähigkeit, ein Problem oder eine Lösung zu erklären, indem man unwichtige Details weglässt. Oder anders ausgedrückt: Es ist die Fähigkeit, eine Idee gedanklich zu formulieren.

Beispiel: Wenn wir ein Fahrrad beschreiben, beziehen wir uns nur auf einige wenige Bestandteile. Wir nennen beispielsweise die Art des Fahrrads und die Farbe. Weitere Details fügen wir nur dann hinzu, wenn sich der Gesprächspartner besonders für Fahrräder interessiert.

Ein Prozess zum Entwickeln von Kompetenzen des informatischen Denkens

Technischer Konstruktionsprozess

Wenn Ingenieure nach einer Lösung für ein Problem suchen, wenden sie einen Konstruktionsprozess an. Sie durchlaufen eine Reihe von Phasen, die sie auf dem Weg zur Lösung anleitet. In den einzelnen Phasen nutzen sie bestimmte Fähigkeiten oder entwickeln diese weiter. Es sind eben diese Fähigkeiten, die wir als „Kompetenzen des informatischen Denkens“ bezeichnen.

Mit WeDo 2.0 durchlaufen die Schülerinnen und Schüler einen ganz ähnlichen Prozess:

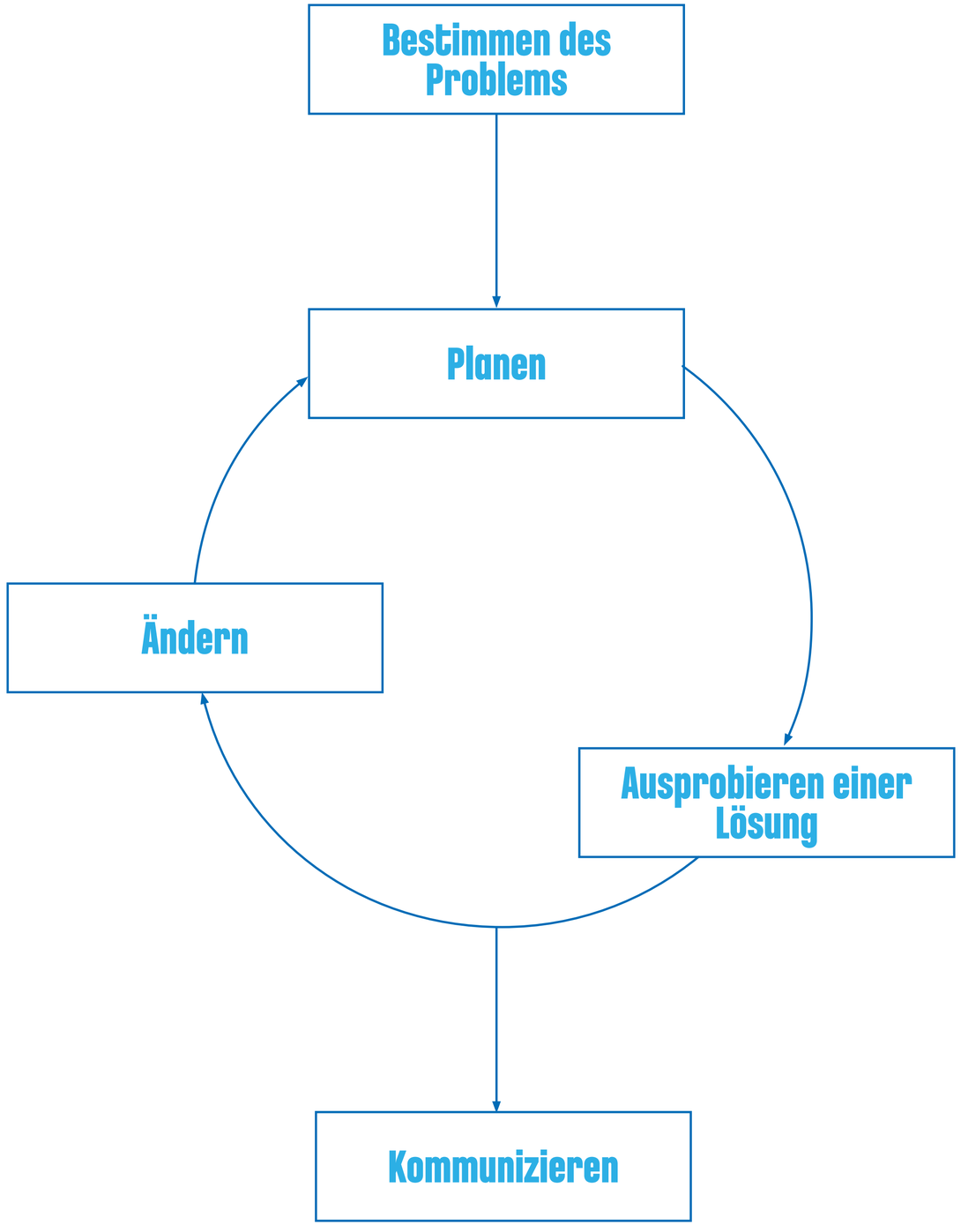
Das Problem bestimmen

Den Schülerinnen und Schülern wird ein Thema vorgestellt, das sie zu einem Problem oder zu einer Situation führt, die sie verbessern möchten. In manchen Fällen können die Probleme viele Facetten haben. Um leichter eine Lösung zu finden, kann das Problem dann in kleinere Teile aufgeteilt werden.

Durch Bestimmen des Problems auf eine einfache Weise und Festlegen einiger Erfolgskriterien erlernen die Schülerinnen und Schüler eine Kompetenz, die wir als „Zerlegung“ bezeichnen.

Anders ausgedrückt:

- *Können die Schülerinnen und Schüler das Problem ohne Hilfestellung erklären?*
- *Können die Schülerinnen und Schüler beschreiben, wie sie am Ende beurteilen, ob sie das Problem erfolgreich gelöst haben oder nicht?*
- *Können die Schülerinnen und Schüler das Problem in kleinere und leichter zu bearbeitende Teile aufteilen?*



Ein Prozess zum Entwickeln von Kompetenzen des informatischen Denkens

Planen

Die Schülerinnen und Schüler sollten sich Zeit dafür nehmen, verschiedene Lösungen für das Problem zu durchdenken. Anschließend fertigen sie einen detaillierten Plan dafür an, wie sie ihre Ideen umsetzen. Sie bestimmen die einzelnen Schritte, die sie abarbeiten müssen, um zur Lösung zu gelangen. Indem sie die Teile der Aufgabe heraussuchen, die sie bereits kennengelernt haben, erlernen sie eine Kompetenz, die wir als „Verallgemeinerung“ bezeichnen.

Anders ausgedrückt: Können die Schülerinnen und Schüler
- *eine Liste von Befehlen erstellen, die programmiert werden sollen?*
- *Teile des Programms bestimmen, die sie verwenden könnten?*
- *Teile des Programms wiederverwenden?*

Ausprobieren

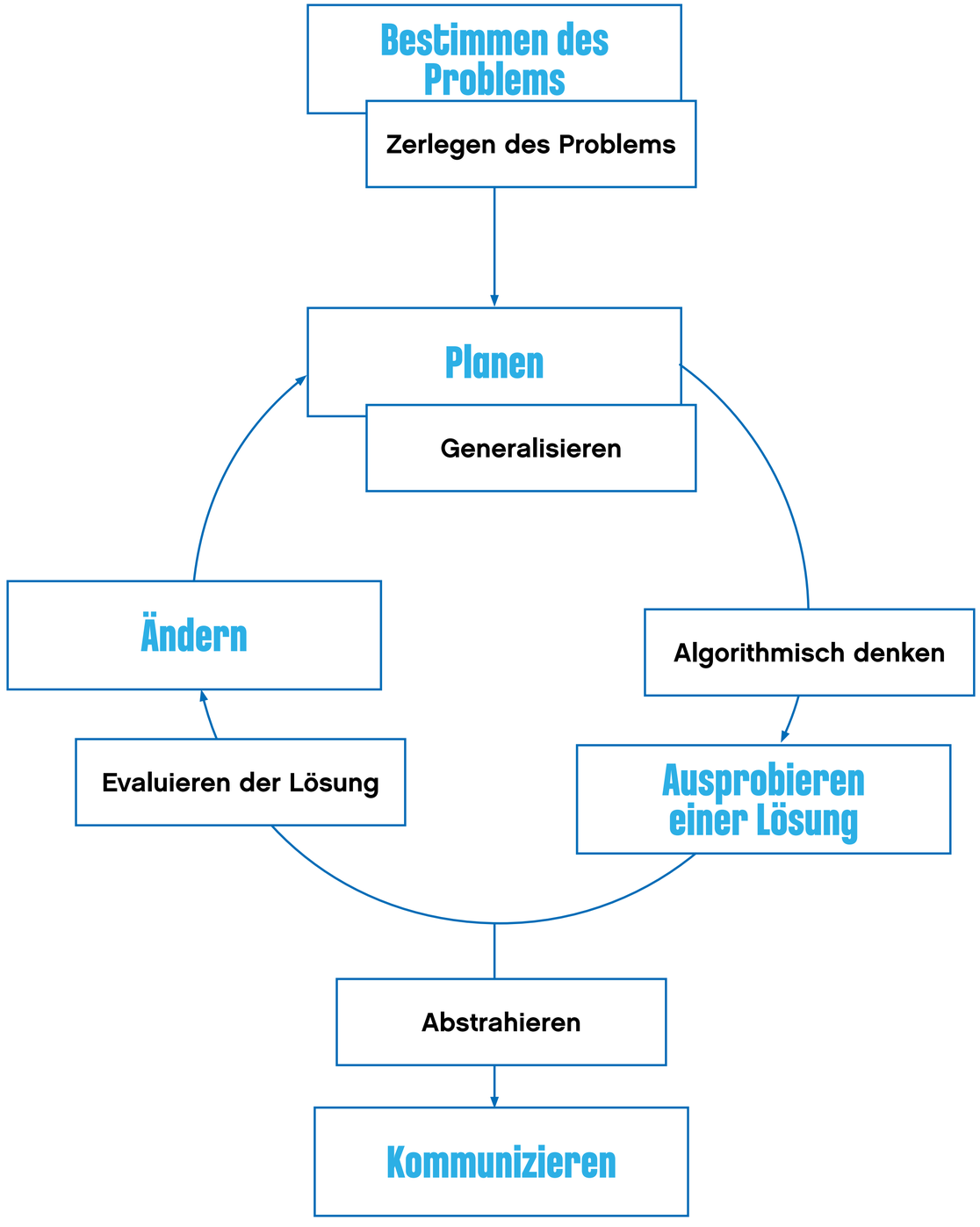
Die Schülerinnen und Schüler sollen dann die finale Version ihrer Lösungen erstellen. In dieser Phase des Prozesses nutzen sie eine symbolbasierte Programmiersprache, um ihre LEGO® Modelle zu aktivieren. Dabei entwickeln die Schülerinnen und Schüler Kompetenzen des Programmierens.

Anders ausgedrückt: Können die Schülerinnen und Schüler
- *eine Lösung im Programm programmieren?*
- *Abfolgen, Wiederholungen, bedingte Anweisungen usw. einsetzen?*

Ändern

Die Schülerinnen und Schüler beurteilen ihre Lösungen und bestimmen dazu, ob ihre Programme und Modelle die Erfolgskriterien erfüllen. Mithilfe dieser Beurteilung stellen sie fest, ob Teile des Programms verändert, repariert oder verbessert und Fehler behoben werden müssen.

Anders ausgedrückt: Können die Schülerinnen und Schüler
- *das Programm wiederholt ausführen?*
- *Probleme im Programm beheben?*
- *beurteilen, ob eine Verbindung zwischen der Lösung und dem Problem besteht?*



Ein Prozess zum Entwickeln von Kompetenzen des informatischen Denkens

Kommunizieren
Die Schülerinnen und Schüler stellen ihre finale Lösung der Klasse vor und erklären dabei, inwieweit ihre Lösung die Erfolgskriterien erfüllt. Bei ihren Präsentationen müssen sie das richtige Maß an Einzelheiten wiedergeben. Auf diese Weise entwickeln sie Kompetenzen in den Bereichen Abstraktion und Kommunikation.

Anders ausgedrückt:
- *Erklären die Schülerinnen und Schüler den wichtigsten Aspekt ihrer Lösung?*
- *Geben die Schülerinnen und Schüler ausreichend Einzelheiten wieder, um ein besseres Verständnis zu ermöglichen?*
- *Erklären die Schülerinnen und Schüler, inwieweit ihre Lösungen die Erfolgskriterien erfüllen?*

Informatisches Denken durch Programmieren erlernen

Um das Programmieren zu erlernen, erhalten die Schülerinnen und Schüler eine Einführung in einige Grundprinzipien des Programmierens. Beim Entwickeln ihrer Lösungen ordnen sie eine Reihe von Befehlen und Strukturen an, mit deren Hilfe sie ihre Modelle zum Leben erwecken.

Die Grundprinzipien, die bei WeDo 2.0 am häufigsten zum Einsatz kommen, sind:

1. Ausgabe
Die Ausgabe wird über das Programm gesteuert, das die Schülerinnen und Schüler schreiben. Beispiele für Ausgaben, die bei WeDo 2.0 vorkommen, sind: Klänge, Lichtsignale, Anzeigen sowie das Ein- und Ausschalten von Motoren.

2. Eingabe
Eingabe bezeichnet die Informationen, die ein Computer oder anderes Gerät empfängt. Die Eingabe kann mithilfe von Sensoren in Form eines Zahlen- oder Textwerts erfolgen. Beispielsweise wandelt ein Sensor, der etwas erkennt oder misst (z. B. einen Abstand), diese Informationen in ein digitales Eingangssignal um, sodass sie im Programm verwendet werden können.

3. Ereignis (Warten auf)
Die Schülerinnen und Schüler können ihren Programmen befehlen, auf ein Ereignis zu warten, bevor eine Abfolge von Befehlen ausgeführt wird. Die Programme können eine vorgegebene Zeit lang warten oder darauf warten, dass der Sensor etwas erkennt.

4. Wiederholung
Die Schülerinnen und Schüler können das Programm so gestalten, dass Aktionen entweder unendlich oft oder über eine vorgegebene Zeit hinweg wiederholt werden.

5. Funktionen
Funktionen sind eine Gruppe von Befehlen, die in bestimmten Situationen zusammen verwendet werden sollen. Zum Beispiel: Eine Gruppe von Blöcken, die eine Lampe blinken lassen, könnte als „Blinkfunktion“ bezeichnet werden.

6. Bedingungen
Die Schülerinnen und Schüler verwenden Bedingungen, um Befehle zu programmieren, die nur dann ausgeführt werden sollen, wenn bestimmte Voraussetzungen erfüllt sind. Bedingungen innerhalb eines Programms zu erstellen, bedeutet auch, dass ein Teil des Programms niemals ausgeführt wird, wenn die Voraussetzungen dafür zu keinem Zeitpunkt erfüllt werden. Zum Beispiel: Wenn sich der Neigungssensor nach links neigt, wird der Motor gestartet, und wenn er sich nach rechts neigt, wird der Motor gestoppt. Wenn sich dann der Neigungssensor niemals nach links neigt, wird auch der Motor niemals gestartet. Und wenn er sich niemals nach rechts neigt, wird auch der Motor niemals gestoppt.